



Arduino LED Exploration

Written By: Steven Robert Cypherd



PARTS:

- [Arduino microcontroller \(1\)](#)

SUMMARY

My program shows you all of the data on your Led from the first light to the normal operation light. From that data you can pick an RS that is good light, but lower in current. You must build the circuit to run my program. You must open a terminal window when you run my program. Bright led can waste a lot of current. Using a higher RS like 470 to 1k you can save almost half of the current. You can set LED_VCC to where the Led is running. Aref is the processor's VCC running my program. You can also set LED_START_VOLTS and LED_MAX_VOLTS for the Led. You can set information in the output for your Led.

RS is found $(V_{in} - V_{led}) / I_{led}$ as $(6 - 1.7) / .02 = 215$ or about 220 ohms.

Currently my program needs to run on a 5 volt system. You can use LED_VCC for any voltage and get to RS for the Test Led. I have a 3.3 volt system now and I will work on my program for it.

I got my Arduino Uno at Makezine.com Store with the Arduino Starter Pack that came with the MakerShield for it. If you have not gotten into Microprocessor Processor Boards you should try the Arduino. There is a ton of support at Arduino.cc/en/. The Arduino IDE(Integrated Development Environment) easy to get started with and can do a lot of stuff

with the processors. You do need to know C. I must say that the C language is uniquely Arduino, but that is the fun of it. They have tons of help and examples.

I was playing with the Light Dependent Resistor (LDR) that came in the Arduino starter pack and I saw it jump when I pasted it over a lit LED. So I built my program to show how Leds work. The Arduinos are wonderful and I love C. This is what processor boards do in our world. They sit in a box collecting data that they format and transmit to another system. My program tests the Led and outputs the data to the terminal.

My program uses Pulse Width Modulation(PWM) on a digital pin to slowly increase the voltage applied to the test LED. I read the LDR for the light output of the Led. Analogue outputs are 256 steps ground to VCC as PWM. Analogue inputs are 1024 steps ground to VCC(Aref). A high enough range to get about four decimal places of accuracy. I created AdcOneVolt by dividing 1024 by Aref(5). I read the volts at the Test Led and R1(220) and I know my volts In so ohms law for current is $\text{VoltsIn} - \text{VoltsLed} / R1$. This was simple with the Arduino.

The lightGap or the difference in the light output by the Led at each Step in the test gave me a plateau test. If the lightGap is low for 3 times in a row it is a plateau. As a semiconductor junction reaches its Saturation point the light, voltage and the current level off in the form of a plateau. My program looks for the first plateau of the Led and ends the test at that point. This is the normal operating voltage and current for the Led at VCC. I also give the Led's first light. The Current and the Input Voltage do change but the Led voltage and its light output do not change much. Increase your RS for each Led and you will have more power for other things.

My program looks more complicated than it is because it is a terminal program. All of that text and logic to control it is what fills up my two loops. Always label the closing brace of a loop when you first start it so that you and everyone else will know what it is. Comments let everyone understand your code. Naming variables so that they connect to what they are make your code easier to understand. Defines do not use any memory. Floating point uses a lot of memory, but they print faster and nicer without any other formatting.

To test Infrared diodes and detectors you need a matched set. The Nano-Meter frequency needs to be close enough for them to work together. The detector goes where the LDR is. You may not need to changes anything in the program.

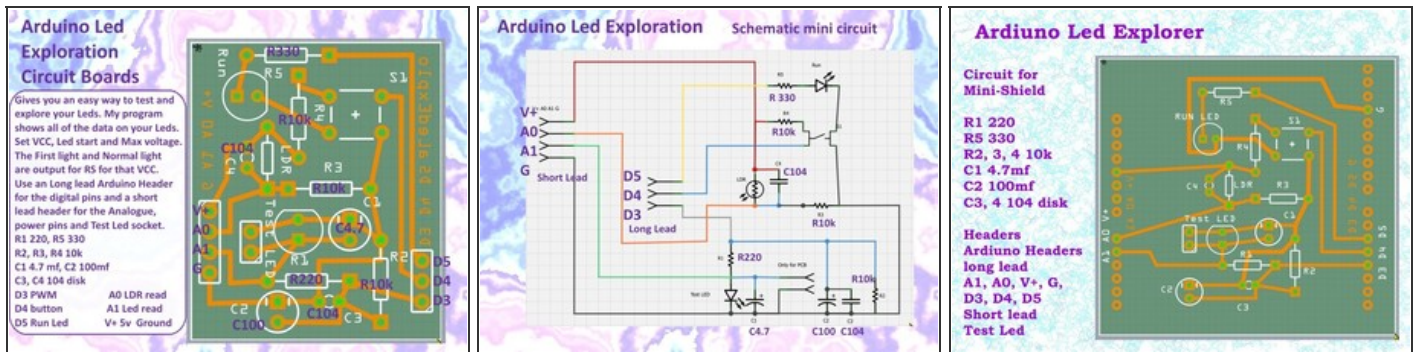
The probe I made out of the simple little LDR that came in the starter pack. I insulated the leads of the LDR and twisted them together to form a cord. I added a bit more wire and

mounted it on a short pin snap-off header. You need to cover the edge and the back of the LDR to block excess light from hitting it. I used black heat-shrink tubing that just fit the LDR to do this. I filled the back of the LDR and the wires with hot-melt glue and added a smaller piece of heat-shrink tubing to make it look nice. I put another piece of heat-shrink tubing on the LDR to form a hood to block more light. Another piece of un-shrunk heat-shrink tubing that just fits the LDR goes over your test LED and then your probe goes into that and touches the test LED. Look at the pictures.

I just created a circuit for this article. You need the circuit to run my program. I created a very small cute circuit board [LedExplore2s circuit.pdf](#) and a mini-shield [LedExplore1mShield.pdf](#). I also made a board with twelve circuits on it [LedExplore2s12.pdf](#). To make the board print the pdf file at 100%. For lasers printers print on the smooth side of the transparency sheet. Make sure you can read the name and port numbers. These boards are the bottom trace of the circuit board. Cut out the circuit on the board's outline.

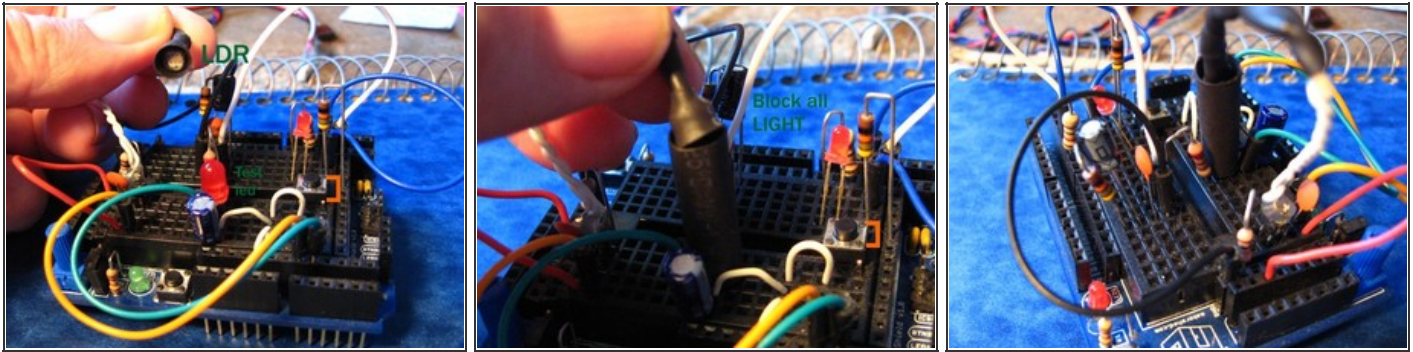
In the circuit the capacitors are necessary for good analogue readings. PWM is noisy. C2-100mf & C3-104 input PWM. C1-4.7mf output test Led voltage read. C4-104 LDR read. R2 10k stabilizes PWM output by providing a load to ground. The LDR and the switch are standard circuits. See the schematic.

Step 1 — Arduino LED Exploration



- R_S is found by $(V_{in} - V_{LED}) / I_{LED}$ as $(6 - 1.7) / .02 = 215$ or about 220.
- My new circuits.

Step 2



- Arduino BreadBoard

Step 3

<p>Arduino Led Exploration Circuit & New Probe</p> <p>The probe is layers of heat shrink tubing. When soldering Leds, transistors, diodes and LDRs keep the leads long to keep heat away from the parts. Form a hood over the LDR to block light. The wires are long enough to test full length lead Leds.</p>	<p>Arduino Led Exploration page 1</p> <pre> /* Arduino LED Exploration 8-15-2011 Modified 11-30-20012 Fixed error: only could do 5 volt Leds Added Aref processor analogue reference Written by Steven B. Cyphard You must open a Serial Monitor after you Upload this program. This is Led Exploration using an Arduino Uno. I read an LDR to to show how LDRs work. The Capacitors are needed to get stable readings. Pulse Width Modulation is noisy. RS is based on the formula RS = (Vin - Vled) / Iled where Vin is 5V5_VCC from the Radiohack Engineer's Mini-Notebook "Optoelectronics Circuits" by Forrest M. Mims III(1984). The Circuit: D3 PWM 10k to Gnd, 100nf cap 15v to Gnd, 10nf cap to Gnd to 220r to Test LED- to A1 to 4.7nf cap 15v to Gnd. Test LED- to Gnd. A0 to LDR 1 to 10k Gnd to 10nf cap to LDR 2 to VCC. D4 Button 1-2 to 10k Vcc, 3-4 to Gnd D5 to R1k to Run Led +, Run Led - to Gnd See my drawing for the circuit. I made the probe from the mini LDR that came in the starter pack. Use black heat shrink tubing to block light and shield the sides and back of the LDR. */ </pre>	<p>Arduino Led Exploration page 2</p> <pre> //User Settings #define ArefC 5 //Processor analogue reference #define ADC_STEPS 1024 //Arduino Uno ADC Steps #define LED_SYS 5 //VCC of the Led and where it will run #define H01_PWM 220 //Resistor between input PWM and the test LED #define LED_MAX_VOLTS 5 //Max LED volts #define LED_START_VOLTS 1 //LED starting volts #define TEST_STEPS 5 //In ADC volts #define PRINT_DETAILS 1 //All Details of the tests #define PRINT_INSTRUCTIONS 0 #define INPUT_DETAILS 0 //Input, Output port values #define INPUT_VALUES 0 //Displays the Actual Reading from the Ports #define LED_DETAILS 1 //Put your details in the print statements //Digital Pins #define TEST_PWM 3 #define SWY_WIDE 4 #define RUN_LED 5 //Analog Pins #define LDR_READ_A0 //LDR Light Dependent Resistor #define LDR_READ_A1 //Test LED float Voltated; //Floats are needed for easy calculations float LedVcc; //and display float VoltFirstLed; //First Light float VoltLastLed; float AmpsFirstLed; float AmpsLastLed; float FirstRS; float AmpsLed; </pre>
---	---	---

- I just created a circuit for this article. You need the circuit to run my program. I created a very small cute circuit board [LedExplore2s.pdf](#) and [circuit.pdf](#) and a mini-shield [LedExplore1mShield.pdf](#).
- I also made a board with twelve circuits on it [LedExplore2s12.pdf](#).
- The program

Step 4

<p>Arduino Led Exploration page 3</p> <pre>float Voltain; float R3ohms; int adcOneVolt; //One volt in adc steps int ledStartVolts; //Led starting volts int ledMaxVolts; //Max for the Led int ledRead; //Analogue read from Test Led int ldrRead; //Analogue read of LDR int ldrLastRead; //For calculation of light gap int lightGap; //Difference of ldrRead between steps int lightDpLow; //When the lightgap drops below 10 int inVolts; int cnt2; int cnt3; int tmp1; int btnState; boolean testRunning; boolean testDone; boolean testStopped;</pre> <pre>void setup() { adcOneVolt = ADC_STEPS / Aref; //create ADC one Volt in an Int ledStartVolts = adcOneVolt * LED_START_VOLTS; ledMaxVolts = adcOneVolt * LED_MAX_VOLTS; ledVcc = (float)LED_VTS; pinMode(RUN_LED, OUTPUT); pinMode(TEST_PWM, OUTPUT); pinMode(BTN_MODE, INPUT); Serial.begin(9600); // End Setup</pre>	<p>Arduino Led Exploration page 4</p> <pre>//Main loop clears each test for the next void loop() { digitalWrite(TEST_PWM, LOW); //Shut off PWM digitalWrite(RUN_LED, LOW); // Turn off test Led delay(20); AmpsFirstLed = 0; ldrLastRead = 0; lightDpLow = 0; cnt2 = 0; testRunning = false; testDone = false; testStopped = false; if (PRINT_INSTRUCTIONS == 1) { Serial.println("Set up the circuit as shown"); Serial.println("Place led to test into the circuit"); Serial.println("Place black heat shrink tubing over the led"); Serial.println("Place your Probe into the tubing and touching "); Serial.println("the Led."); Serial.println(""); } if (INPUT_DETAILS == 1) { Serial.print("PWM port "); Serial.print(TEST_PWM, DEC); Serial.print(" LDR port "); Serial.print(LDR_READ, DEC); Serial.print(" Test LED port "); Serial.print(RUN_LED, DEC); Serial.println(""); } Continued</pre>	<p>Arduino Led Exploration page 5</p> <pre>Continue Serial.print("Led Starting Volts "); Serial.print((float)ledStartVolts / adcOneVolt); Serial.print(" Max "); Serial.print((float)ledMaxVolts / adcOneVolt); Serial.print(" VCC "); Serial.print(" "); Serial.print(" LedVcc "); Serial.print(" RL "); Serial.println(" "); Serial.println(" "); Serial.println(""); //Report data goes here if (LED_DETAILS == 1) { Serial.println(""); Serial.println("Test Green Led Date 10/29/12"); } Serial.println("Please place the LED in the test position"); Serial.println("Press the Button to Start or Stop the Test"); //Wait for user to set up the test do { tmp1 = digitalRead(BTN_MODE); //Pauses until Button is pushed while(tmp1 == HIGH); btnState = tmp1; //0 is Button Down. Serial.println("Running Test "); digitalWrite(RUN_LED, HIGH); cnt3 = 0;</pre>
---	---	---

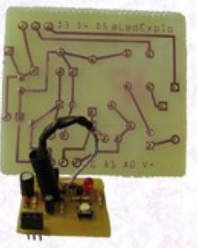
- The program

Step 5

<p>Arduino Led Exploration page 6</p> <pre>//Testing Loop----- for (inVolts = ledStartVolts; inVolts <= ledMaxVolts; inVolts = TEST_STEPS) { analogWrite(TEST_PWM, inVolts / 4); //PWM voltage to Test LED delay(20); ldrRead = analogRead(LDR_READ); //Read Light output of Led with LDR delay(20); ledRead = analogRead(LED_READ); //Read Led volta delay(5); //Wait for light from the led before printing details if (ldrRead < 10) { //Lowest light reading needed Serial.print("-"); cnt2 = 1; if (cnt2 > 35) { //Page width Serial.println("-"); cnt2 = 0; } continue; } if (testRunning == false) { Serial.println(""); testRunning = true; } //Trying to clear capacitors digitalWrite(TEST_PWM, LOW); //Shut off PWM</pre>	<p>Arduino Led Exploration page 7</p> <pre>//Main calculations cnt3 = 1; VoltLastLed = (float)ledRead / adcOneVolt; VoltIn = (float)inVolts / adcOneVolt; AmpsLed = (VoltIn - VoltLastLed) / R01_PWM; Rohms = (ledVcc - VoltLastLed) / AmpsLed; AmpsLed = AmpsLed * 1000; //Adjust for mA display if (AmpsFirstLed == 0) { AmpsFirstLed = AmpsLed; VoltFirstLed = VoltLastLed; FirstRS = R3ohms; } if (ldrLastRead < ledRead) { //Setup Last reading for LightGap LightGap = ledRead - ldrLastRead; //Positive Lightgap only } else { LightGap = ldrLastRead - ledRead; } //Check test button tmp1 = digitalRead(BTN_MODE); if (tmp1 == false && btnState == true) { testDone = true; testStopped = true; } else { btnState = tmp1; }</pre>	<p>Arduino Led Exploration page 8</p> <pre>if (lightGap <= 10) { //Lowest of Light Gap lightDpLow = 1; //Three or more is a big drop in light if (lightDpLow == 3) testDone = true; //Number of lowest light gap to end the test } else { if (lightDpLow != 0) lightDpLow = 0; } if (testDone == true) { if (testStopped == true) Serial.println("**** Test Stopped ****"); Serial.println("Test Results "); Serial.println("For VCC of "); Serial.print(ledVcc); Serial.println(" volts"); Serial.print("First light volts is "); Serial.print(VoltFirstLed); Serial.print(" "); Serial.print("AmpsFirstLed "); Serial.print("mA "); Serial.println("Normal volts is "); Serial.print(VoltLastLed); Serial.print(" "); Serial.print("AmpsLastLed "); Serial.print("mA "); Serial.print("RS "); Serial.println("ohms, 0 "); Serial.println(""); break; //End the test here }</pre>
--	--	--

- My program

Step 6

<p>Arduino Led Exploration page 9</p> <pre>if (PRINT_DETAILS == 1) { if (INPUT_VALUES == 1) { Serial.print("Input Values in "); Serial.print(" Led "); Serial.print(ledRead); Serial.print(" LDR "); Serial.print(ldrRead); Serial.println(""); } else { Serial.print(cnt3, DEC); Serial.print(" IN "); Serial.print(VoltIn); Serial.print(" Led V "); Serial.print(VoltLastLed); Serial.print(" "); Serial.print(AmpsLed); Serial.print("mA, RS "); Serial.print(R3ohms, 0); Serial.print(" Loop "); Serial.print(lightGap); Serial.print(" LED "); Serial.println(ldrRead); } }</pre>	<p>Arduino Led Exploration page 10</p> <pre>if (lightDpLow == 0) { //Catches last readings VoltLastLed = VoltLastLed; //Test ends AmpsLastLed = AmpsLed; //No more reads } ldrLastRead = ldrRead; //For... The testing Loop ----- if (testDone == true) { Serial.println("Test is Done"); Serial.println("-----"); Serial.println(""); Serial.println(""); } else { Serial.println(""); Serial.println("*****Test failed*****"); Serial.println("Check Led and voltage settings "); Serial.println(""); } } //Loop Set-up the Test and wait. -----</pre>	<p>Arduino Led Exploration Circuit Board</p> <p>To make my program easier to use I made a mini circuit board. I used Fritzing.com and created a 4x6 PCB with 12 of my circuits on it. They are all drilled and sanded.</p> <p>If you would like one of my boards send me a self-addressed stamped envelope and a check or money order for 5 dollars US payable to me and I will send you a board.</p> <p>Steven R. Cypher 1018 N. Hayworth Av. #6 West Hollywood Ca, 90046 I only have 11 of my boards.</p> <p>If you are a club or a group you can just make my boards</p> <p>Use the article for the part values and placement. I am not going build anymore than one of my boards so I can fix my program. My original program only tested 5 volt Leds. I fixed that by adding Aref for the processor VCC. I also cleaned up the final output. Please let me know how you like my program and my circuit board.</p> 
---	---	--

- My program

Step 7

The diagram illustrates Ohm's Law and its application in a circuit. On the left, a triangle contains the variables V (Voltage), I (Current), and R (Resistance). To its right are the formulas: $V = I \times R$, $I = V / R$, and $R = V / I$. Below this is a circuit diagram showing a voltage source V_{in} connected in series with a resistor R_S and an LED labeled Led , which is connected to ground G . The calculation for the resistor value is shown as: $R_S = (V_{in} - V_{Led}) / I_{led}$ and $R_S = (6 - 1.7) / .02 = 215$.

On the right, a circular diagram (Ohm's Law wheel) shows the relationships between P (Power), I (Current), and E (Voltage). The center contains P, I, and E. The wheel is divided into segments with the following formulas: E^2/R , P/E , E/R , $\sqrt{P/R}$, E/I , P/I^2 , $I \times R$, E^2/P , $\sqrt{P \times R}$, $I \times E$, $I^2 \times R$, and P/I .

- Ohms Law

Build the circuit and down load the source code from Letsmakerobots.com under something else.

This document was last generated on 2012-12-05 12:52:29 AM.